8. Дубинин В.Н., Дроздов Д.Н. Проектирование и реализация систем управления дискретными событийными системами на основе иерархических модульных недетерминированных автоматов (Ч. 1. Формальная модель). Известия высших учебных заведений. Поволжский регион. Технические науки. – 2016. – № 1 (37). – С. 28-39.

9. Моделирование распределенных многокомпонентных программных систем и их тестирование на основе автоматных вероятностных моделей /С.М. Старолетов. – Барнаул: Изд-во АлтГТУ, 2011. – 107 с.

10. Мартин Фаулер Предметно-ориентированные языки программирования: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 576 с.

11. Н.П.Вашкевич Недетерминированные автоматы в проектировании систем параллельной обработки [Текст]/Н.П.Вашкевич: учебное пособие. – Пенза: Изд-во Пенз.гос.ун-та, 2004. – 280 с.

12. Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. – СПб.: Наука и Техника, 2006. – 320 с.

УДК 004.451

АНАЛИЗ НАИБОЛЕЕ ПЕРСПЕКТИВНЫХ АЛГОРИТМОВ ПЛАНИРОВАНИЯ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

А.И. Мартышкин
кандидат технических наук, доцент, доцент кафедры вычислительных машин и систем, ФГБОУ ВО «Пензенский государственный технологический университет», г. Пенза, Россия, e-mail: Alexey314@yandex.ru

Аннотация. В статье рассмотрены основные принципы работы алгоритмов планирования операционных систем реального времени. Выявлено, что задачи в операционных системах реального времени делятся на периодические и апериодические и для их планирования используются статические и динамические алгоритмы планирования.

Ключевые слова: операционная система, алгоритмы планирования, планировщик, приоритет, эффективность, производительность, система реального времени.

ANALYSIS OF THE MOST PROMISING ALGORITHMS FOR PLANNING REAL-TIME OPERATING SYSTEMS

A.I. Martyshkin
Ph.D., Associate Professor, Associate Professor of the Department of Computers and Systems, FGBOU VO 'Penza State Technological University', Penza, Russia, e-mail: Alexey314@yandex.ru

Abstract. The article describes the basic principles of real-time operating system planning algorithms. It is revealed that tasks in real-time operating systems are divided into periodic and aperiodic, and static and dynamic scheduling algorithms are used for their planning.

Keywords: operating system, scheduling algorithms, scheduler, priority, efficiency, performance, real-time system.

Введение. Алгоритм планирования представляет собой неотъемлемую часть любой операционной системы, во многом определяющую эффективность использования аппаратных ресурсов встраиваемой вычислительной системы. Однако, для разных классов систем критерии эффективности организации вычислений различны. Специфика систем реального времени определяется требованием своевременного выполнения прикладных задач. Исследование алгоритмов планирования является актуальной темой, поскольку алгоритм планирования – это часть программного обеспечения, используемого в системах реального времени.

Цель работы. Цель работы – провести анализ и сравнение наиболее известных алгоритмов планирования операционных систем реального времени.

Материал и результаты исследований. В настоящее время для решения задач эффективного планирования в операционных системах реального времени существует два основных подхода: статические алгоритмы планирования; динамические алгоритмы планирования. Основной отличительной чертой приведенных выше подходов является время определения приоритетов задач, находящихся в системе. В статических алгоритмах приоритет определяется до запуска планировщика, а в динамических порядок выполнения задач определяются в ходе работы системы [1]. В статье представлен анализ существующих алгоритмов планирования операционных систем реального времени, рассмотрен принцип работы, а также достоинства и недостатки того или иного алгоритма планирования.

Алгоритмы для планирования периодических задач. Статический алгоритм планирования. Rate Monotonic (RM). В статических алгоритмах планирования приоритеты задач определяется на этапе их создания, т.е. перед запуском планировщика системы и на протяжении всей работы системы не меняются. Планировщик принимает решение об установке определенного приоритета задаче на основе введённых пользователем различных временных характеристик, таких как: период исполнения, время исполнения и другие. Суть алгоритма RM заключается в присвоении статических приоритетов задачам на основе их периодов. Приоритеты задачам назначаются следующим образом – задача с наименьшим приоритетом получает самый высокий приоритет[4]. Алгоритм RM может быть использован, только при условии, что все задачи в системе отвечают следующим требованиям.

1 все задачи независимы между собой;

2 каждая задача должна быть завершена в течение своего периода;

3 каждая задача может быть приостановлена более высокоприоритетной задачей;

4 время выполнения каждой задачи постоянно;

5 прерывание процесса происходит мгновенно [2].

Данный алгоритм является стабильным, т.к. при перегрузке процессора высокоприоритетные задачи продолжают исполняться вовремя. Ключевым параметром для алгоритмов планирования является загрузка (1), являющаяся суммой отношений времени выполнения задач к периоду задач.

$$U = \sum_{i=0}^{n} \frac{c_i}{p_i} \quad (1)$$

Для проверки на возможность планирования с помощью алгоритма RM определенного набора задач, необходимо выполнение неравенства (2).

$$\sum_{i=0}^{n} \frac{c_i}{p_i} \leq n * (2^{\frac{1}{n}} - 1) \quad (2)$$

n – количество задач в системе;

c_i – время выполнения задачи;

p_i – период выполнения задачи.

Для алгоритма RM при количестве задач, стремящемся к бесконечности, максимальная загрузка процессора равна 69%. При использовании процессора меньше чем на 69%, алгоритм RM гарантирует успешное выполнение задач в системе, если же уровень использования процессора превышает 69%, то набор задач может быть спланирован, но выполнение всех задач не гарантировано [3].

На рисунке 1 представлен пример работы системы, где присутствует три периодических задачи со следующими временными характеристиками и значением загрузки, находящейся в левой части неравенства.

$$0{,}68 \leq 3 * \left(2^{\frac{1}{3}} - 1\right)$$

$$0{,}68 \leq 0{,}77$$

Таблица 1 – Временные характеристики задач системы

Задача	Время выполнения	Период
1	10	30
2	10	40
3	5	50

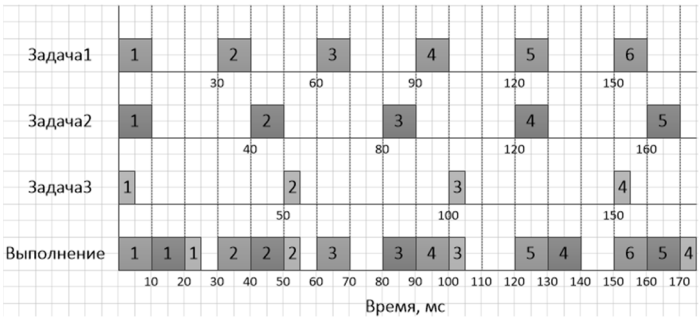

Рисунок 1 – Пример работы алгоритма RM

Как видно из рисунка 1 в системе имеется три задачи с различными периодами исполнения, в соответствии с которыми им назначаются приоритеты: задача 1 имеет высший приоритет, задача 2 – средний и задача 3 – самый низкий приоритет. На протяжении всего времени работы системы задача 1 вытесняет все задачи, выполняющиеся в момент её возникновения, задача 2 вытесняет только задачу 3 и во время, когда ни одна задача не выполняется, процессорное время переходит к задаче 3.

Динамические алгоритмы планирования. В динамических алгоритмах планирования приоритеты задачам назначаются во время работы системы на основе анализа текущей ситуации. Планировщик во время работы системы отдает приоритет задачам, удовлетворяющим различным условиям, в зависимости от алгоритма планирования. Динамические алгоритмы бывают двух типов: для периодических и для апериодических задач. Периодическая задача – задача, требующая обработки или активизирующаяся через определенный интервал времени, называемый периодом задачи. Апериодическая задача – задача, активизация которой происходит при наступлении определенного события или состояния в системе.

Earliest Deadline First (EDF). Суть алгоритма заключается в том, что в момент события планирования, планировщиком выбирается задача, которая имеет самый ранний крайний срок, т.е. время, оставшееся до конца периода. Основными событиями планирования операционных систем реального времени являются – готовность задачи к выполнению и приостановка выполнения задачи [4]. Формальным условием данного алгоритма является то, что максимальная загрузка процессора должна быть меньше 100%. Имеет преимущества над статическим алгоритмом при больших загрузках системы. Этот алгоритм является не стабильным, поскольку и низкоприоритетная, и высокоприоритетная задачи могут получить процессорное время при перегрузке [5]. На рисунке 2 представлен пример работы системы, в которой присутствует три периодических задачи со следующими временными характеристиками и значением загрузки = 97,5 %

Таблица 2 – Временные характеристики задач системы

Задача	Время выполнения	Период
1	15	30
2	15	40
3	5	50

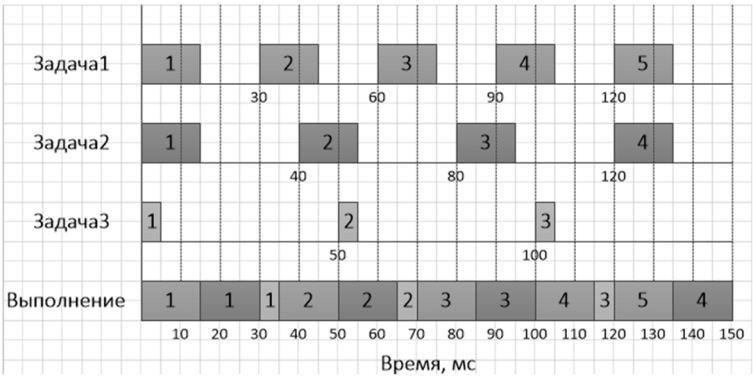

Рисунок 2 – Пример работы алгоритма EDF

Least Laxity First (LLF). Данный алгоритм основан на следующем принципе – в момент наступления события планирования наивысший приоритет получает задача с наименьшим резервом времени. Резервом времени называется разность между временем до крайнего срока и оставшимся временем выполнения задачи. Как и в алгоритме EDF, формальным условием данного алгоритма является то, что максимальная загрузка процессора должна быть меньше 100%.

Также, как и алгоритм EDF, данный алгоритм является нестабильным при перегрузке процессора. На рисунке 3 представлен пример работы системы, в которой присутствует три периодических задачи со следующими временными характеристиками и значением загрузки = 91 %.

Таблица 3 – Временные характеристики задач системы

Задача	Время выполнения	Период
1	10	30
2	15	40
3	10	50

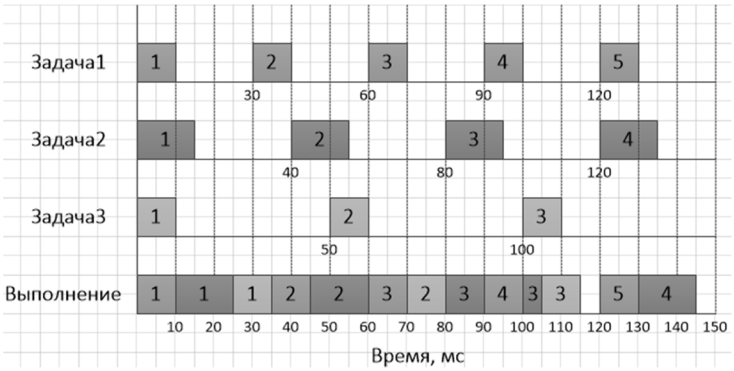

Рисунок 3 – Пример работы алгоритма LLF

Алгоритмы для планирования апериодических задач. Background. В основе алгоритма Background лежит фоновое выполнение апериодических задач, т.е. периодические задачи системы обслуживаются на основе любого просто алгоритма, например, RM, а в оставшихся окнах процессорное время выделяется для апериодических задач, которые готовы к выполнению в данный момент времени. Апериодические задачи обслуживаются, когда все периодические задачи приостановлены. В случае, если апериодическая задача не успела завершить свое выполнение, т.е. её вытеснили, то она возвращается в конец списка готовых апериодических задач.

Основным недостатком данного алгоритма является то, что он не позволяет работать с апериодическими задачами реального времени.

На рисунке 4 представлен пример работы системы, в которой присутствует две периодических задачи и два апериодических события, возникающих в определенные временные интервалы, со следующими временными характеристиками и значением загрузки периодических задач, находящейся в левой части неравенства.

$$0{,}75 \leq 2 * \left(2^{\frac{1}{2}} - 1\right)$$

$$0{,}75 \leq 0{,}83$$

Таблица 4 – Временные характеристики периодических задач системы

Задача	Время выполнения	Период
1	10	20
2	10	40

Таблица 5 – Временные характеристики апериодических событий системы

Событие	Время выполнения	Время возникновения
1	10	40
2	10	50

Рисунок 4 – Пример работы алгоритма Background

Deferrable Server. Алгоритм Deferrable Server (сервер, допускающий задержку) основан на алгоритме RM. В системе создается периодическая задача сервер, которая предназначается для обслуживания апериодических запросов и имеет такой параметр как бюджет. По количеству бюджета планировщик определяет, будет ли обрабатываться в данном периоде сервера апериодический запрос. Если значение бюджета равно нулю, то управление получают готовые периодические задачи до момента обновления бюджета сервера, т.е. до начала следующего периода сервера [6].

На следующем рисунке представлен пример работы системы, в которой присутствует две периодические задачи, периодический сервер с бюджетом равным 5 и апериодическое событие, со следующими временными характеристиками и значением загрузки периодических задач, находящейся в левой части неравенства:

$$0{,}741 \leq 3 * \left(2^{\frac{1}{3}} - 1\right)$$

$$0{,}741 \leq 0{,}77$$

Таблица 6 – Временные характеристики периодических задач системы

Задача	Время выполнения / Бюджет	Период
1	10	20
2	15	40
сервер	5	30

Таблица 7 – Временные характеристики апериодических событий системы

Событие	Время выполнения	Время возникновения
1	10	40

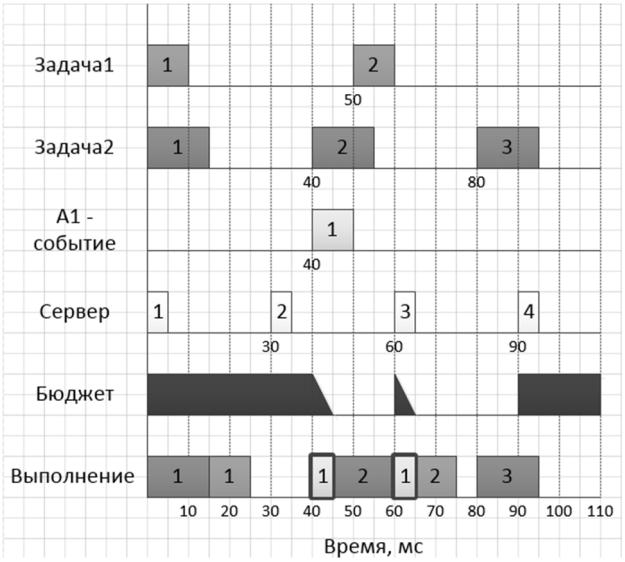

Рисунок 5 – Пример работы алгоритма Deferrable Server

Sporadic Server. Алгоритм Sporadic Server (спорадический сервер) по принципу аналогичен алгоритму сервера, допускающего задержку (deferrable server). Для обслуживания апериодических событий создается специальная периодическая задача(сервер), имеющая максимальный приоритет. Аналогично алгоритму deferrable server, в алгоритме sporadic server у сервера имеется свой бюджет, по истечении которого управление передается следующей задаче из списка готовых к выполнению. Единственное отличие алгоритма sporadic server от deferrable server состоит в том, что бюджет времени пополняется не в каждый период сервера, а через “период пополнения”, т.е. через определенное количество времени, после начала очередного периода сервера [6].

На рисунке 6 представлен пример работы системы, в которой присутствует две периодические задачи, периодический сервер с бюджетом равным 5, “периодом пополнения” равным 5 и апериодическое событие, со следующими временными характеристиками и значением загрузки периодических задач, находящейся в левой части неравенства:

$$0{,}741 \leq 3 * \left(2^{\frac{1}{3}} - 1\right)$$
$$0{,}741 \leq 0{,}77$$

Таблица 8 – Временные характеристики периодических задач системы

Задача	Время выполнения / Бюджет + “Период пополнения”	Период
1	10	20
2	15	40
сервер	5 + 5	30

Таблица 9 – Временные характеристики апериодических событий системы

Событие	Время выполнения	Время возникновения
1	10	40

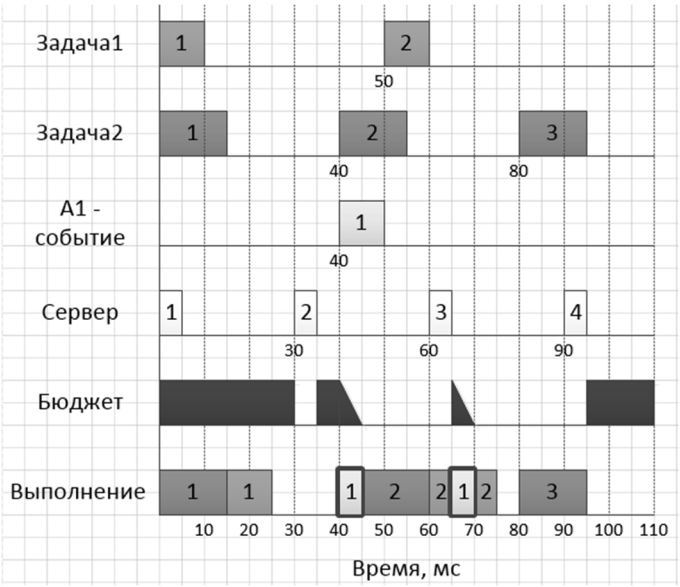

Рисунок 6 – Пример работы алгоритма планирования Sporadic Server

Выводы. В результате проведения обзора разобраны основные принципы работы алгоритмов планирования операционных систем реального времени. Также было выявлено, что задачи в операционных системах реального времени делятся на периодические и апериодические и для их планирования используются статические и динамические алгоритмы планирования.

В статических алгоритмах приоритет определяется до запуска планировщика, а в динамических порядок выполнения задач определяются в ходе работы системы. В связи с этим при реализации представленных в первой главе алгоритмов необходимо предусмотреть ввод и обработку различных временных параметров задач, таких как период, время выполнения. Для алгоритмов, имеющих периодический сервер, необходимо обеспечить ввод параметров – тип задачи (апериодическая, периодическая), период сервера, бюджет сервера, период пополнения.

Работа выполнена при финансовой поддержке стипендии Президента РФ молодым ученым и аспирантам на 2018-2020 гг. (СП-68.2018.5).

ЛИТЕРАТУРА

1. Бурдонов, И. Б. Операционные системы реального времени / И.Б. Бурдонов, А. С. Косачев, В.Н. Пономаренко // Препринт Института системного программирования РАН. – 2006. – №14. – С. 3.

2. Audsley, N. C. Applying new scheduling theory to static priority preemptive scheduling / N.C. Audsley, A. Burns, M. Richardson, K. Tindell, A. Wellings // Software Engineering Journal. – 2002. – Vol. 8. – № 5. – P. 1.

3. Bini, E. The space of rate monotonic schedulability / E. Bini, G.C. Buttazzo // IEEE Real-Time Systems Symposium. – 2002. - №23. – P. 1.

4. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. 4-е изд. – Санкт-Петербург: Питер, 2015. – С. 602.

5. Mok A. Multiprocessor scheduling in a hard real-time environment / A. Mok, M. Dertouzos // Texas Conference on Computing Systems. – 1978. – №7. – P. 3.

6. Buttazzo G.C. Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (Real-Time Systems Series) / G.C. Buttazzo. – 3rd ed. – London: Springer, 2011. – P. 143.

УДК 37.033:504.06

ОСОБЛИВОСТІ ВИКЛАДАННЯ ІНЖЕНЕРНОЇ ТА КОМП'ЮТЕРНОЇ ГРАФІКИ ДЛЯ ЗДОБУВАЧІВ ОСВІТИ ПРИРОДООХОРОННОГО ПРОФІЛЮ У НТУ «ДНІПРОВСЬКА ПОЛІТЕХНІКА»

А.В. Павличенко[1], І.М. Мацюк[2]

[1]доктор технічних наук, завідувач кафедри екології та технологій захисту навколишнього середовища, Національний технічний університет «Дніпровська політехніка», м. Дніпро, Україна, e-mail: pavlichenko.a.v@nmu.one

[2]кандидат технічних наук, доцент кафедри конструювання, технічної естетики і дизайну, Національний технічний університет «Дніпровська політехніка», м Дніпро, Україна, e-mail: matsyukin@ua.fm

Анотація. В статті проаналізовано особливості викладання дисциплін «Інженерна графіка» та «Прикладна комп'ютерна графіка» здобувачам вищої освіти за спеціальністю 183 «Технології захисту навколишнього середовища». Очікувані результати навчання за цими дисциплінами забезпечують формування навичок та вмінь використання комп'ютерної техніки та сучасного програмного забезпечення під час розроблення нових та удосконалення існуючих технологій захисту навколишнього середовища.

Ключові слова: природоохоронна освіта, технології захисту навколишнього середовища, інженерна графіка, прикладна комп'ютерна графіка, системи автоматизованого проектування, природоохоронне обладнання.